普通高等教育“十三五”规划教材

PUTONG GAODENG JIAOYU “13·5” GUIHUA JIAOCAI

吉首大学“十三五”精品教材

编译原理课程辅导

莫礼平　周恺卿　宋海龙　编著

冶金工业出版社
www.cnmip.com.cn

普通高等教育“十三五”规划教材

吉首大学“十三五”精品教材

编译原理课程辅导

莫礼平　周恺卿　宋海龙　编著

北　京
冶 金 工 业 出 版 社
2019

内 容 提 要

本书分“理论”和“实验”两篇编写。“理论篇”依据清华大学王生原、董渊、张素琴等编著的《编译原理（第3版）》的结构和内容为主线编写而成，并对课程体系和教学内容进行了适当优化和补充，主要以程序设计语言编译器设计及实现的基本原理、基本方法和技术为核心，分十章进行编写。“实验篇”主要针对词法分析及语法分析核心算法的五个实验项目，分五章进行编写。

本书针对性强、选题范围广、难易适当，能够为读者熟练掌握编译原理知识、抓住重点、突破难点提供有益的帮助。

本书既可作为高等院校计算机科学与技术、软件工程、网络工程等本科专业编译原理课程的教学参考书，又可作为上述专业学生进行编译原理课程学习和实验的辅导书，也可供参加计算机相关专业硕士研究生复试或计算机技术与软件专业技术资格（水平）考试的读者使用。

图书在版编目(CIP)数据

编译原理课程辅导/莫礼平等编著. —北京：冶金工业出版社，2019.7

普通高等教育“十三五”规划教材

ISBN 978-7-5024-8143-8

Ⅰ.①编… Ⅱ.①莫… Ⅲ.①编译程序—程序设计—高等学校—教材 Ⅳ.①TP314

中国版本图书馆CIP数据核字(2019)第144430号

出 版 人 谭学余

地　　址 北京市东城区嵩祝院北巷39号　邮编 100009　电话 (010)64027926

网　　址 www.cnmip.com.cn　电子信箱 yjcbs@cnmip.com.cn

责任编辑 宋 良　美术编辑 吕欣童　版式设计 孙跃红

责任校对 郑 娟　责任印制 牛晓波

ISBN 978-7-5024-8143-8

冶金工业出版社出版发行；各地新华书店经销；三河市双峰印刷装订有限公司印刷

2019年7月第1版，2019年7月第1次印刷

787mm×1092mm　1/16；18.75印张；450千字；289页

39.00元

冶金工业出版社　投稿电话　(010)64027932　投稿信箱　tougao@cnmip.com.cn

冶金工业出版社营销中心　电话　(010)64044283　传真　(010)64027893

冶金工业出版社天猫旗舰店　yjgycbs.tmall.com

前　言

本书是根据培养新时代大学生对教学提出的新要求，在总结学生学习难点的基础上，根据学生实际需要，精心编著而成的辅助教材，旨在帮助学生正确理解编译系统相关概念和原理，把握重点和难点，掌握解题技巧，进而达到使学生深刻理解程序设计语言的设计及实现原理和技术，真正了解程序设计语言相关理论，在宏观上把握程序设计语言精髓的目标。

书中“理论篇”的十章内容均按知识结构、知识要点、例题分析、习题与习题解答五个部分来编写。知识结构部分，按照清华大学王生原等编著的《编译原理（第3版）》教材中对教学内容的安排，给出了知识结构图。知识要点部分，简明扼要地归纳了各章的主要内容和需要重点掌握的知识点，着重理清其中的概念、原理和方法，将教材中抽象理论、复杂算法以通俗易懂的语言进行解释，为学生理解和掌握课程内容提供指导。例题分析部分，主要针对那些重要原理和算法，特别是针对学生在学习中遇到的重点和疑难问题，以例题形式进行了详尽的分析和解释，并给出了一些简便的解题方法，以帮助学生拓宽思路，加深对课程内容的理解，提高分析问题和解决问题的能力。习题与解答部分，针对各章重点选编了适当数量的各类习题，提供给读者练习，所有习题均给出了参考解答。“实验篇”的五章内容分别针对五个实验项目，按实验指南和实验参考源代码两部分来指导学生进行实验操作。实验指南部分，从实验目的、实验内容、实验要求、运行结果示例、实验提示、总结分析与讨论等方面为学生提供详细的实验指导。实验参考源代码部分，提供了基于C/C++或Java语言编写的源程序代码清单。

编者的初衷，是通过对本书的阅读，能够让读者把握编译原理理论课程和实验课程的主线，加深对基本概念和原理的理解，掌握解题的思路与方法，以及编程实践的技能，帮助读者融会贯通、举一反三，以增强分析问题、解决问题的能力。

本书既是编者多年来从事编译原理课程教学实践的工作积累，也得益于参考了多种编译原理教科书和全国计算机技术与软件专业技术资格（水平）考试历届试题。同时，感谢我们的学生，是他们强烈的求知欲望和勤学好问、认真学习的态度，给了我们编著教材的动力。

由于编者水平所限，书中不妥之处，诚请读者批评指正。

编 者

2019 年 3 月

目　录

理 论 篇

实 验 篇

理论篇

LILUN PIAN

1 引 论

1.1 知识结构

本章的知识结构如图 1-1 所示。

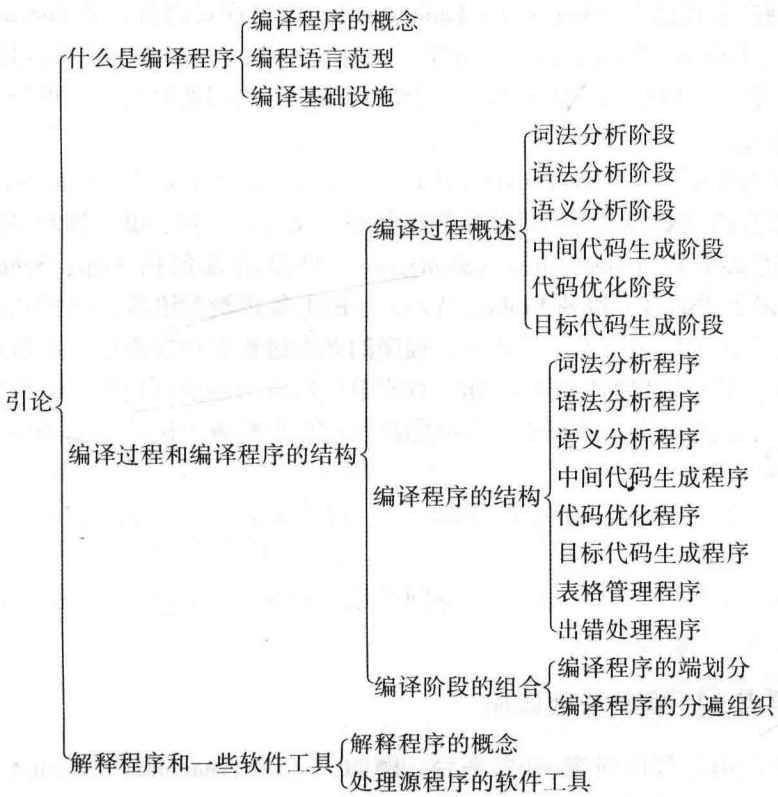

图 1-1 第 1 章知识结构图

1.2 知识要点

本章的知识要点主要包括以下内容：

【知识要点 1】编译程序

(1) 从基本功能来看，编译程序（Compiler）是一种较为复杂的语言翻译程序（Translator）。它通过对源程序进行分析（Analysis），识别源程序的语法结构信息，理解源

程序的语义信息，反馈相应的出错信息，并根据分析结果及目标信息进行综合（Synthesis），生成语义上等价于源程序的目标程序。

（2）传统编译程序的源语言通常为高级语言（例如 Fortran、Algol、C、Pascal、Ada、C++、Java、Lisp、Prolog、Python），目标语言通常为机器级语言（例如汇编语言、机器语言）或较低级的虚拟机语言（例如 Java 虚拟机语言 ByteCode）。

（3）编译程序是现代计算机系统的基本组成部分之一，通常由词法分析程序、语法分析程序、语义分析程序、中间代码生成程序、目标代码生成程序、代码优化程序、表格管理程序和出错处理程序等成分构成。

【知识要点 2】编程语言的主要范型

（1）强制命令式语言（Imperative Languages），即过程式语言，如 Fortran、Algol、Cobol、C、C++、Pascal、Basic、Java、C#等。该型语言中，一个过程可看做是一系列动作，其动作由命令驱动，以语句形式表示，一个语句接一个语句地执行。本课程介绍的编译技术针对这型语言。

（2）面向对象语言（Object-Oriented Languages），如 Smalltalk、Simula6、Java、C++、C#等。该型语言的主要特点是提供抽象数据类型，支持封装性、继承性和多态性。

（3）陈述式语言（Declarative Languages）。该型语言包括 Lisp、Scheme、Haskell、ML、Caml 等函数式语言，以及 Prolog、YACC、BNF 等逻辑型语言。函数式语言即应用式语言。该型语言注重程序所表示的功能，程序的开发过程是从前面已有函数出发构造出更加复杂的函数，对初始数据集进行操作，直到最后形成的函数可以得到最终结果。逻辑型语言即基于产生式的语言。该型语言程序的执行过程是检查一定的使能条件，满足时，则执行适当的动作。

（4）并发语言（Concurrent Languages），如并发 Pascal、Ada、Java、Linda、HPF、OpenMP 等。

（5）其他语言，如 Signal、Lustre 等同步语言（Synchronous Languages），Perl、PHP 等脚本语言（Scripting Languages）。

【知识要点 3】编译基础设施

（1）共享的编译程序研究/开发平台。例如，SUIF（Stanford），Zephyr（Virginia and Princeton），IMPACT，LLVM(UIUC)，GCC（GNU Compiler Collection），Open64（SGI、中科院计算所、Intel、HP、Delaware 等）。

（2）多源语言多目标机体系结构。例如，GCC 有 C、C++、Objective C、Fortran、Ada、Java 等诸多前端，以及支持 30 多类体系结构、上百种平台的后端。

（3）多级别的中间表示。例如，Open64 的中间表示语言 WHIRL 分 5 个级别设计。

【知识要点 4】编译过程

一般编译程序的工作过程按阶段进行，每个阶段将源程序从一种表示形式转换成另一种表示形式。典型的阶段划分方法是将整个编译过程分为如下六个阶段：

（1）词法分析。该阶段的任务是对构成源程序的字符串进行扫描和分解，扫描源程序

字符流，识别出有词法意义的单词，返回单词的类别和单词的值，或词法错误信息。所谓单词，是指逻辑上紧密相连的一组字符。这些字符具有集体含义，比如，标识符是由字母字符开头，后跟字母、数字字符的字符序列组成的一种单词，用于表示变量名、常量名、函数名、过程名等。此外还有保留字（关键字或基本字）、常量、运算符、界符等单词。该阶段输入构成源程序的字符串，输出单词符号序列。

（2）语法分析。该阶段的任务是根据语言的语法产生式，对单词符号串（符号序列）进行语法分析，识别出各类语法短语（能够表示成语法树的语法单位），判断输入串在语法上是否正确。该阶段输入单词序列，输出可表示成语法树的语法单位的单词序列。

（3）语义分析。该阶段的任务是按语义产生式，对语法分析所得到的语法单位进行语义分析，审查有无语义错误，为代码生成阶段收集类型信息，并进行类型审查以及违背语言规范的报错处理。该阶段输入语法分析后的单词序列，输出语义分析后带语义信息的单词序列。

（4）中间代码生成（并非所有的编译程序都包含此阶段）。该阶段的任务是将语义分析得到的源程序，变成一种结构简单、含义明确、易生成、易翻译成目标代码的内在代码形式。常用的中间代码是形如“（算符，运算对象 1，运算对象 2，结果）”的 TAC 代码（即三地址指令代码，又称四元式）。该阶段输入语义分析后的单词序列，输出中间代码。

（5）代码优化（也可以放到目标代码生成阶段之后）。该阶段的任务是对中间代码或目标代码进行等价的变换改造等优化处理，使生成的代码更高效。该阶段输入中间代码或目标代码，输出优化后的中间代码或目标代码。

（6）目标代码生成。该阶段的任务是将语义分析后的单词序列或中间代码生成特定机器上的绝对或可重定位的指令代码或汇编指令代码。该阶段输入语义分析后的单词序列或优化后的中间代码，输出特定机器上的目标代码。

【知识要点 5】编译程序结构

（1）编译程序的构成部分。编译过程的六个阶段的任务，再加上表格管理和出错处理的工作，可分成几个模块或程序完成，分别称为词法分析程序、语法分析程序、语义分析程序、中间代码生成程序、代码优化程序、目标代码生成程序、表格管理程序和出错处理程序。其中，出错处理包括检查错误并报告出错信息，以及排除错误并恢复编译工作。

（2）编译程序结构图。一个典型的编译程序结构框图如图 1-2 所示。

【知识要点 6】编译阶段的组合与分遍

（1）编译程序中与源语言相关而与目标机无关的部分（第 1~4 阶段）称为编译前端，与目标机相关而与源语言无关的部分（第 6 阶段）称为编译后端。第 5 阶段置前端或后端都可以。基于前后端的编译程序组合方式如下：

①同一源语言的编译前端+不同后端=不同机器上同一源语言的编译程序。

②不同源语言的编译前端生成同一种中间语言+共同后端=同一机器上不同语言的编译程序。

（2）对源程序或源程序的中间结果从头到尾扫描一次称为一遍。每一遍扫视完成一个或几个阶段的工作。一个编译程序可由一遍或多遍完成。

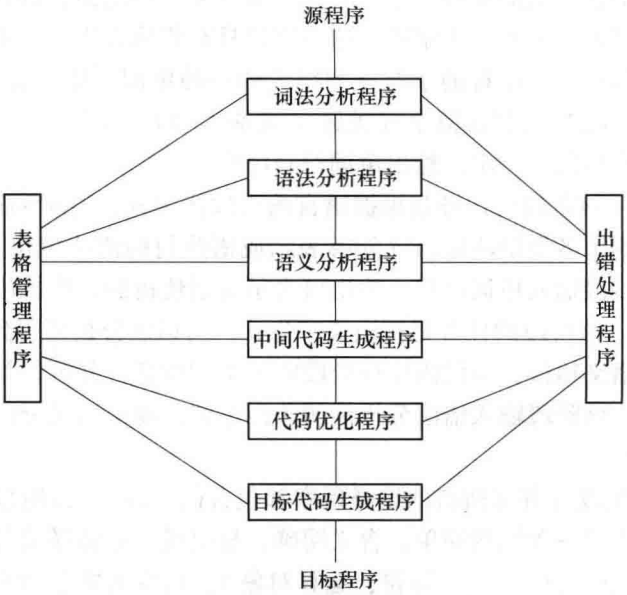

图 1-2 典型的编译程序结构框图

（3）编译阶段前后端的组合、实际编译程序分遍的主要参考因素都是源语言与目标机器的特征。

【知识要点 7】解释程序

（1）解释程序的概念。在计算机上运行高级语言的程序主要有两个途径：一是把该程序翻译为这个计算机的指令代码序列，这就是编译过程；二是编写一个程序，它解释所遇到的高级语言程序中的语句并且完成这些语句的动作，这样的程序就是解释程序。

（2）解释程序的特点。解释程序不产生目标程序文件，不区别翻译阶段和执行阶段，翻译源程序的每条语句后直接执行，程序执行期间一直有解释程序守候，常用于实现虚拟机。从功能上说，解释程序能够让计算机直接执行高级语言。它每遇到一个语句，就要对这个语句进行分析以决定语句的含义，并执行相应的动作，不需要生成目标代码。

【知识要点 8】编译方式与解释方式的区别

（1）编译方式对应的源语言是高级语言，目标语言是低级语言（汇编或机器语言）的翻译程序；解释方式接受所输入的源语言程序后，不生成目标代码，就直接解释执行源程序。

（2）基于解释执行的程序可以动态修改自身；而基于编译执行的程序则需要动态编译技术，难度较大。

（3）解释方式有利于人机交互。

（4）解释方式的执行速度通常要比编译方式慢。

（5）编译产生的目标代码所占用的空间通常要多于解释方式。

（6）编译方式与解释方式的根本区别在于是否生成目标代码。

【知识要点 9】兼有编译程序和解释程序的语言

（1）BASIC、LISP、PASCAL 等语言，既有编译程序，又有解释程序。Java 语言的处理环境也是既有编译程序，又有解释程序。

（2）Java 编译器把 Java 代码翻译成独立于机器的 Java “字节代码”。运行时，目标装置中的校验器分析这些字节代码，以确保代码的安全执行。在目标装置中，内置一个 JVM（Java 虚拟机）。该虚拟机用一个解释器或一个 JIT（适时）编译器把字节代码翻译成目标处理器能够识别的机器语言代码。

1.3 例题分析

【例题 1-1】 写出编译程序对如下 C 语言源程序片段进行处理时六个编译阶段的返回结果。

$Sum = Data_1 + Data_2 * 100;$

分析与解答：

（1）词法分析。该阶段的任务是对构成源程序的字符串进行扫描和分解，扫描源程序字符流，识别出有词法意义的单词，返回单词的类别和单词的值，或词法错误信息。所谓单词，是指逻辑上紧密相连的一组字符。这些字符具有集体含义，比如，标识符是由字母字符开头，后跟字母、数字字符的字符序列组成的一种单词，用于表示变量名、常量名、函数名、过程名等。此外，还有保留字（关键字或基本字）、常量、运算符、界符等单词。该阶段输入构成源程序的字符串，输出单词符号序列。本题中的 C 语言源程序片段经词法分析后返回：

标识符 Sum
运算符 =
标识符 $Data_1$
运算符 +
标识符 $Data_2$
运算符 *
常数 100
界符 ;

（2）语法分析阶段。该阶段的任务是在词法分析的基础上将单词序列分解成各类语法短语，如“程序”“语句”“表达式”等等。一般这种语法短语也称语法单位，可表示成语法树。语法分析的功能依据语法产生式（即描述程序结构的产生式）进行层次分析，把源程序的单词序列组成语法短语（表示成语法树），确定整个输入串是否构成一个语法上正确的程序。经语法分析得知，上述 C 语言源程序片段中的单词序列是 C 语言的“赋值语句”，可表示成图 1-3 所示的三叉树或图 1-4 所示的二叉树语法树形式。

（3）语义分析阶段。该阶段的任务是审查语法上正确的源程序有无语义错误。例如，源程序中有些语法成分（比如，使用了没有声明的变量，或给一个过程名赋值，或调用函

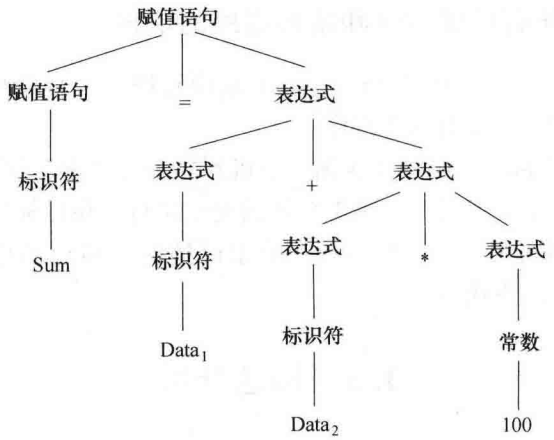

图 1-3 赋值语句 $Sum=Data_1+Data_2*100$ 的三叉语法树

数时参数类型不合适，或者参加运算的两个变量类型不匹配等），按照语法产生式去判断，它是正确的，但它不符合语义产生式。这些都属于语义错误。题中 C 语言源程序片段语法正确，在编译程序进行语义分析阶段的类型审查之后，会将整型量转换为实型量。语义分析的结果可以体现在语法分析所得到的分析树上。在语法树上增加一个单目运算符结点（结点的名称为 inttoreal），表示进行将整型量变成实型量的语义处理。那么，语义分析后的语法树变成图 1-5 所示结果。

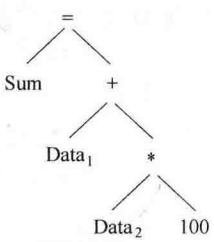

图 1-4 赋值语句 $Sum=Data_1+Data_2*100$ 的二叉语法树

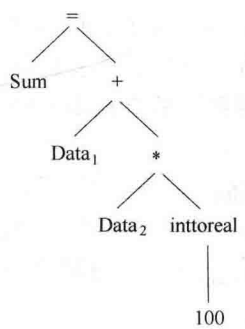

图 1-5 赋值语句 $Sum=Data_1+Data_2*100$ 语义分析后的二叉语法树

（4）中间代码生成阶段。在语义分析之后，大部分编译程序将源程序变成一种称为中间语言或中间代码的内部表示形式。中间代码是一种结构简单、含义明确的记号系统。这种记号系统可以设计为多种多样的形式，重要的设计原则有两点：一是容易生成；二是容易将它翻译成目标代码。很多编译程序采用了一种近似“三地址指令”的“TAC”中间代码，形式为：（运算符，运算对象 1，运算对象 2，结果）。题中 C 语言源程序片段转换为如下所示的四个 TAC 形式的中间代码：

$(inttoreal,100,\_,t_1)$

$(*,Data_2,t_1,t_2)$

$(+,Data_1,t_2,t_3)$

$(=,t_3,\_,Sum)$

（5）代码优化阶段。该阶段的任务是对前一阶段产生的中间代码，通过删除公共子表达式、强度削弱、循环优化等优化技术进行等价变换或改造，使生成的目标代码更为高效，既省时间又省空间。通常，代码优化工作会降低编译程序的编译速度，因此，编译优化阶段常常作为编译过程的可选阶段，编译程序具有控制机制以允许用户在编译速度和目标代码的质量间进行权衡。本题中，可以先进行将 100 转换成实型数的代码优化掉；同时，因生成 t_3 的 TAC 只起到传值作用，也可以被优化掉。故上述中间代码优化后变换为如下的两个 TAC：

$(*, Data_2, 100.0, t_1)$
$(+, Data_1, t_1, Sum)$

（6）目标代码生成阶段：该阶段的任务是把中间代码变换成特定机器上的绝对指令代码或可重定位的指令代码或汇编指令代码。这是编译的最后阶段，它的工作与硬件系统结构和指令含义有关，涉及硬件系统功能部件的运用、机器指令的选择、各种数据类型变量的存储空间分配以及寄存器和后缓寄存器的调度等。本题中，可使用两个寄存器（R_1 和 R_2）将优化后的中间代码生成如下形式的某种汇编代码：

```
MOVF   Data2   R2
MOLF   #100.0  R2
MOVF   Data1   R1
ADDF   R1   R2
MOVF   R1   Sum
```

上面的第一条指令将 $Data_2$ 的内容送至寄存器 R_2，第二条指令将 R_2 中的值与实常数 100.0 相乘（这里用#表示将 100.0 作为常数处理，第三条指令将 $Data_1$ 移至寄存器 R_1，第四条指令将 R_1 加上前面计算得到的 R_2 中的值，第五条指令将寄存器 R_1 的值移至 Sum 对应的地址单元中。

1.4 习 题

1-1 解释程序与编译程序都是一种语言_____程序。两者的区别在于：前者接受源程序后立即运行源程序，不生成_____，直接就输出结果。BASIC、LISP、PASCAL 等语言，既有编译程序，又有解释程序。Java 语言的处理环境也是既有编译程序，又有解释程序。Java _____器把 Java 代码翻译成独立于机器的 Java “字节代码”。

1-2 编译过程六个阶段的任务分别由六个子程序模块来完成。一个完整的编译程序还应包括_____管理和_____处理程序。因此，在典型编译程序框图中，从一个源程序翻译成目标程序，要涉及_____个子程序模块的调用。

1-3 编译阶段按_____可分为编译前端和编译后端。其中，与目标机有关的阶段一般属于_____，而与源语言相关的阶段一般属于_____。

1-4 典型高级程序设计语言编译系统的工作过程常分为六个阶段，即词法分析、语法分析、语义分析、中间代码生成、_____、目标代码生成。编译阶段的两种组合方式是_____组合法和按遍组合法，进行组合的主要参考因素都是源语言和